

**-- CROSS REFERENCE TO RELATED APPLICATIONS**

*A1*  
This application claims priority of U.S. Provisional Application Serial No. 60/254,837 entitled "System And Method For Hardware Assisted Spinlock", filed December 12, 2000, which is hereby incorporated by reference in its entirety.--

**IN THE CLAIMS:**

Please amend claims 6, 7, 18 and 19 as follows:

*A1*  
6. (Amended) The multiple processor computer system of claim 5 wherein the queue comprises a plurality of entries, each entry comprising a first data field for containing a processor identification, a second data field for containing a lock request time, and a third data field for containing a priority of a request for a lock.

*A2*  
7. (Amended) The multiple processor computer system of claim 6 wherein each request for a lock on the shared resource has a corresponding entry in the queue, and wherein the crossbar structure determines which request is granted the lock based at least in part upon the respective priorities contained in the third data fields of the queue entries for each request.

*A3*  
18. (Amended) The crossbar structure of claim 17 wherein the queue comprises a plurality of entries, each entry comprising a first data field for containing a processor identification, a second data field for containing a lock request time, and a third data field for containing a priority of a request for a lock.

*A3*  
19. (Amended) The crossbar structure of claim 18 wherein each request for a lock on the shared resource has a corresponding entry in the queue, and wherein the crossbar structure determines which request is granted the lock based at least in part upon the respective priorities contained in

*AB*  
the third data fields of the queue entries for each request.

*JW(B)*  
Please add the following new claims 48-52:

*AB*  
48. (Newly added) The multiple processor computer system of claim 7 wherein if the third data fields of more than one of the queue entries contains the same and highest priority, the crossbar structure further determines which request is granted the lock based upon the respective lock request times contained in the second data fields of those queue entries.

*AB*  
49. (Newly added) The crossbar structure of claim 19 wherein if the third data fields of more than one of the queue entries contains the same and highest priority, the crossbar structure further determines which request is granted the lock based upon the respective lock request times contained in the second data fields of those queue entries.

*AB*  
50. (Newly added) The method of claim 24 wherein determining a processor further comprises:  
reading at least one priority field and a processor identification corresponding to each priority field, from the queue;  
determining that more than one priority field contains the same and highest priority;  
reading a time field corresponding to each priority field containing the same and highest priority;  
determining a processor based on the processor identification corresponding to the earliest time in the time fields; and  
granting the lock to the determined processor corresponding to the earliest time in the time fields.

51. (Newly added) The method of claim 36 wherein determining a processor further comprises:

- reading at least one priority field and a processor identification corresponding to each priority field, from the queue;
- determining that more than one priority field contains the same and highest priority;
- reading a time field corresponding to each priority field containing the same and highest priority;
- determining a processor based on the processor identification corresponding the earliest time in the time fields; and
- granting the lock to the determined processor corresponding to the earliest time in the time fields.

52. (Newly added) The method of claim 46 wherein determining a processor further comprises:

- reading at least one priority field and a processor identification corresponding to each priority field, from the queue;
- determining that more than one priority field contains the same and highest priority;
- reading a time field corresponding to each priority field containing the same and highest priority;
- determining a processor based on the processor identification corresponding the earliest time in the time fields; and
- granting the lock to the determined processor corresponding to the earliest time in the time fields.